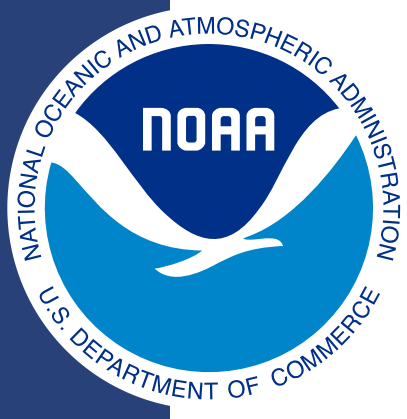


NOAA
OMD

OMD OFFICE NOTE 006

https://doi.org/10.25923/frz0-h186

Toward Development of Machine Learning Version of Real-Time Mesoscale Analysis (RTMA)

Miodrag Rancic, Manuel De Pondeca, R. James Purser, Annette Gibbs, Edward Colon, Matthew Morris, Gang Zhao, Guoqing Ge, Terra Ladwig

Any opinions or views expressed or implied herein are those of the authors and do not necessarily reflect the views of NOAA or the Department of Commerce.

This manuscript was produced with the assistance of Generative AI. All AI contributions have been subject to human review, including full quality assurance and extensive editing.



US DEPARTMENT OF COMMERCE
National Oceanic and Atmospheric Administration
National Weather Service
Office of Modeling and Development
Riverdale Park, MD
August, 2026

Toward Development of Machine Learning Version of Real-Time Mesoscale Analysis (RTMA)

Miodrag Rancic[1] (https://orcid.org/0000-0002-8288-9581)
Manuel De Pondeca[1] (https://orcid.org/0000-0002-9793-2650)
R. James Purser[1] (https://orcid.org/0000-0003-4670-4129)
Annette Gibbs[1] (https://orcid.org/0009-0000-4234-4571)
Edward Colon (https://orcid.org/0009-0005-9438-2787)
Matthew Morris[2] (https://orcid.org/0000-0003-0204-8625)
Gang Zhao[2] (https://orcid.org/0000-0003-3574-4117)
Guoqing Ge[3] (https://orcid.org/0000-0002-9934-0608)
Terra Ladwig[3] (https://orcid.org/0000-0002-0950-274X)

[1] Lynker Technologies at NOAA National Weather Service, Office of Modeling and Development, College Park, MD, USA
[2] Science Applications International Corporation at NOAA National Weather Service, College Park, MD, USA
[3] Global Systems Laboratory at NOAA Office of Oceanic and Atmospheric Research, Boulder, Colorado, USA

Abstract

The Real-Time Mesoscale Analysis (RTMA) project provides near-real-time analyses for the contiguous United States, Alaska, Guam, Puerto Rico, and Hawaii with a high horizontal spatial resolution between 1.25 and 3 km, with future extensions planned up to 1 km. The analyses are issued hourly with plans for sub-hourly or 'on-demand' being considered. High computational efficiency is essential for the success of this project. Recent advances in artificial intelligence (AI) and machine learning (ML) have led to significant improvements in atmospheric modeling, resulting in a more efficient production of high-quality weather products, such as real-time forecasts and analyses. This text contains a description of the preliminary exploration of various ideas and options available to begin work on the development of an ML version of the RTMA system, which can be used for high-resolution weather analysis. The paper also outlines several strategic directions, presents initial prototypes, and proposes a conceptual roadmap for future development.

Office of Modeling and Development Office Note 006
August, 2026

This page is intentionally left blank

Table of Contents

1 Introduction 4

2 General strategy 5

3 Application of a diffusion model for data assimilation within RTMA 6

4 Adaptation of an end-to-end approach 8

5 ML version of 4DVAR approach 8

6 Examples 9
- 6.1 Inclusion of physics constraints in ML emulators 10
- 6.2 First ML prototype of data analysis for 2D RTMA 12
- 6.3 Second ML prototype of data analysis for 2D RTMA 13
- 6.4 PINNs-assisted 4DVAR 15

7 Conclusions 17

8 References 22

APPENDICES

A Glossary of acronyms used 25

1. Introduction

The rapid development and widespread application of Artificial Intelligence and Machine Learning (AI/ML) in meteorology and atmospheric sciences in general are primarily driven by two key factors

- A surge in novel atmospheric observations and the accumulation of diverse data sources such as reanalysis datasets and model outputs;
- The advancement and implementation of Graphics Processing Units (GPUs), which offer exceptionally fast processing of massive parallel datasets, making them much more efficient for deep learning and neural networks training than traditional central processing units (CPUs; e.g., Gyawali, 2023).

Machine learning emulators of weather forecasting models have gained considerable recognition for their efficiency and strong predictive performance. Notable examples include

- For global scales
  - GraphCast (Lam et al., 2023),
  - FourCastNet (Pathak et al., 2022),
  - Pangu-Weather (Bi et al., 2023).
- For meso- and storm-scales
  - StormCast (Pathak et al., 2024),
  - ML LAM (Oskarsson et al., 2023),
  - HRRRCast (Abdi et al., 2025),
  - Pathak et al. (2026).
- End-to-end ML forecasting systems.
  - Aardvark Weather (Vaughan et al., 2024),
  - Weather Mesh (Du et al., 2025).
  - OCELOT (Gholoubi et al., 2026)

Although AI/ML has been applied to various components of data assimilation — such as observation impact assessment (e.g., Virmani et al., 2024), estimation and correction of model error (e.g., Bonavita and Laloyaux, 2020) and quality control (e.g., Sun et al., 2021; DesRosiers and Bell, 2024) — its use as a full emulator of a complete DA system is still in the relatively early stages of development.

NOAA's Real-Time Mesoscale Analysis (RTMA, e.g., De Pondeca et al. 2011, and Ge et al., 2024) is a high temporal and spatial resolution analysis system developed and maintained by the Office of Modeling and Development (OMD) of the National Weather Service (NWS) and the Global Systems Laboratory (GSL) of the Office of Oceanic and Atmospheric Research (OAR). It provides a near-real-time hourly analysis for the contiguous territory of the United States (CONUS), Alaska, Guam, Puerto Rico and Hawaii at spatial resolutions between 1.25 and 3 km. Future developments aim to achieve temporal and spatial resolutions of up to 5 min and 1 km, respectively, including 4D paradigm such as on-demand analyses at any requested time. Currently, the RTMA is primarily used to support National Digital Forecast Database operations at the National Weather

Service (NWS) as well as the aviation operations of the Federal Aviation Administration (FAA). A prerequisite for the continued development of the RTMA is the attainment of increasingly higher computational efficiency.

Therefore, we started looking for an appropriate method that can serve as an AI/ML version of RTMA, capable of efficiently learning from the background field and combined effects of various observations interacting among themselves and surface conditions and take advantage of historical records of Unrestricted Mesoscale Analysis (URMA) and reanalysis as training datasets. This text will summarize some of the main candidates for this task and share some lessons learned about the general application of AI/ML in data assimilation (DA), ending with a presentation of several simple prototypes and their results and a suggestion of a tentative road map for the long-term development.

This text continues with a description of the general strategy in Section 2. Section 3 describes diffusion models as a long-term goal that can bring probabilistic capability and access to very high resolutions. Section 4 considers an end-to-end approach, and Section 5 introduces a possible adaptation of ML 4D-Var as a method to aid in better capturing atmospheric dynamics during data assimilation. Section 6 introduces several simple prototypes illustrating the advantages of some of the discussed techniques, along with a suggestion that imposing physical constraints could be beneficial within data assimilation architectures. Section 7 concludes with a discussion and a draft of a possible roadmap for development.

2. General strategy

We envision that the development of a machine learning-based RTMA system (ML RTMA[1]) will be designed, at least in the initial stage, to complement, rather than fully replace, traditional data assimilation approaches. Advanced methods such as a four-dimensional variational (4DVAR) approach, hybridized with the Ensemble Kalman Filter will continue to play a critical role in providing high-quality reference data for *offline* training of the ML model. Meanwhile, ML RTMA will operate *online*, providing a rapid analysis suitable for real-time applications. Such a dual approach will ensure both high quality and efficiency in the production of mesoscale analyses.

We also assume that the 6-hour time-lagged, quality controlled version of RTMA, the Unrestricted Mesoscale Analysis (URMA), which benefits from the assimilation of additional observations that arrive too late for the RTMA, will continue to be updated so that it could be used for testing and retraining of the ML RTMA (among its other objectives). In addition, at one stage we will start using an ML forecasting model for the production of background fields, so that the final product will consist of an end-to-end application of continuous and fast ML machinery for the computation of high-quality real-time mesoscale analyses.

Once the initial version of ML RTMA has been prototyped, we will continue searching for additional sources of high-quality training fields for parameters such as cloud ceiling height and visibility, which are difficult to analyze in the current variational-based RTMA/URMA system. Such fields will be incorporated into the ML RTMA retraining at a later time using Transfer Machine Learning (TML)[1] methods (see, e.g., Zhuang et al., 2021), where a previously trained model is adapted and fine-tuned on a new, often smaller dataset. We expect that such an approach would lead to a higher quality of the analyses produced, just as the use of the NCEP reanalysis and

[1]Although the expressions AI and ML are often used interchangeably, we will keep the terminology 'ML RTMA', being more specific for this project.

[1]The standard terminology in machine learning is Transfer Learning (TL). We adopt TML here to avoid confusion with Tangent Linear (TL) models in data assimilation.

the Global Data Assimilation System (GDAS) in the retraining of the ML global forecasting model GraphCast (Sadeghi Tabas et al., 2025) substantially improved the quality of its global forecasts.

We think that a sound strategy for the development of ML RTMA encompasses an incremental approach, where the focus in the first stage will be on the development of an ML 2D RTMA, followed by the development of the full 3D variant. The developed ML data assimilation system(s) should first be tested over a smaller domain, such as the state of Oklahoma, which has an extremely high-quality network of atmospheric and environmental observations, the Oklahoma Mesonet (Brock et al., 1995; McPherson et al., 2007).

In the rest of this text we will review several major candidate approaches for the task, though, as always when dealing with an extremely fast developing field, this list is by no means complete and does not exclude other possible candidates, which were not included at the time of writing this text.

3. Application of a diffusion model for data assimilation within RTMA

Generative diffusion models operate by iteratively transforming random noise into structured data through a learned denoising process. At each step, the model refines an initial noisy sample, gradually reconstructing complex data distributions such as images, text, or atmospheric fields. This process is guided by patterns learned from large datasets, allowing the model to generate highly realistic outputs that capture the variability and spatial structure of the target domain. The generative capability of diffusion models makes them especially powerful for data synthesis, data augmentation, and reconstruction of missing information. Figure 1 schematically illustrates this process: gradually adding Gaussian noise to an image during the forward diffusion process produces an unstructured noisy pattern. Reversing this process and applying conditioning then guides the denoising steps toward generating new, physically consistent samples.

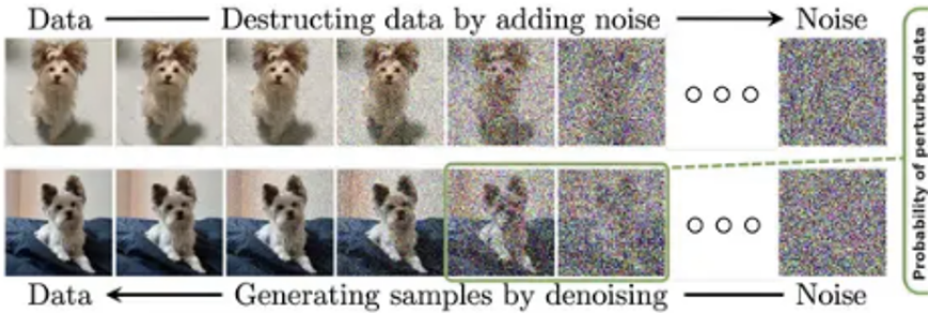


Figure 1: Schematic illustration of diffusion process. (Adopted from Yang et al., 2025)

Useful references can be found online at, for example:

- "What are Diffusion Models?"
  https://lilianweng.github.io/posts/2021-07-11-diffusion-models/
- "How diffusion models work: the math from scratch."
  https://theaisummer.com/diffusion-models/
- "Diffusion Models: A comprehensive survey of methods and applications."
  https://arxiv.org/pdf/2209.00796.pdf

Comprehensive overviews of diffusion theory and implementations can be found in, e.g., Weng (2021) and Yang et al. (2023).

ML data assimilation based on the denoising stage of a diffusion model (DiffDA), introduced by Huang et al. (2024), takes advantage of the structural similarity between weather prediction models and denoising diffusion frameworks. DiffDA utilizes the pre-trained global ML model GraphCast as the core of its diffusion process. During the denoising stage, conditioning methods (Rombach et al., 2022) developed for stable diffusion guide the process toward the observed state. Sparse or irregular observations are mapped onto the model grid using spatial interpolation and soft masking—where continuous weighting replaces binary selection to reflect observation density and measurement uncertainty. This allows the conditioning mechanism to implicitly act as the observation operator without requiring explicit $H(\mathbf{x})$ matrix inversions or adjoint calculations. Consequently, sparse observations are smoothly integrated into high-resolution gridded representations, steering the denoising trajectory toward the target analysis state. This approach aligns directly with the objectives of transitioning to an ML-based RTMA system.

This approach effectively leverages the learned representation of atmospheric dynamics from the pre-trained ML backbone, making the assimilation process computationally efficient and accurate. Once DiffDA is fully trained, its primary advantage over traditional variational or ensemble methods is its superior execution speed during the inference stage. The conditioning mechanism allows flexible integration of diverse and sparse observations, enabling new observations to be incorporated dynamically at inference time without model retraining. Furthermore, the framework is intrinsically robust against missing data, as the diffusion process infers unobserved variables based on learned atmospheric correlations. When paired with ML forecasting models, this approach paves the way for a fully data-driven, end-to-end ML pipeline.

Various mechanisms exist to control the relative weights assigned to observations during the denoising stage. For example, the attention mechanism—particularly multi-head self-attention (Vaswani et al., 2017)—enables the network to dynamically focus on relevant spatial regions during the generative process. In diffusion architectures, attention layers are commonly integrated within U-Net backbones[2], allowing the model to weigh the importance of specific spatial regions or observational inputs at each denoising step. This selective focus improves the model's ability to capture long-range spatial dependencies and contextual cues, ultimately enhancing the physical coherence of the generated fields. By dynamically adjusting attention weights, the system flexibly balances information from background predictions, observation constraints, and prior states.

Additional applications of denoising methods to data assimilation can be found in recent studies by Selmer et al. (2024), Manshausen et al. (2025), Shen (2025), and Hodyss and Morzfeld (2025).

For implementation within ML RTMA, we propose utilizing a regional ML model, such as ML-LAM (Oskarsson et al., 2023), which has already been configured and partially trained over the contiguous United States (CONUS) domain at a 3-km horizontal resolution. Additionally, Diffusion-LAM (Larsson et al., 2025) offers a direct, regional diffusion architecture suitable for this purpose. Beyond computational efficiency, the denoising approach introduces a much-needed probabilistic component to the RTMA framework. By formulating data assimilation as a reverse stochastic process, denoising acts as a generative engine that delivers probabilistic information: rather than producing a single deterministic analysis, it yields a probability distribution of plausible atmospheric states. Over observation-sparse regions, a probabilistic denoiser—such as DiffDA or the WindSR architecture (Ma et al., 2025)—fills observational gaps with multiple physically plausible realizations.

[2]A convolutional neural network architecture characterized by a symmetrical, U-shaped structure that combines a contracting path to capture multi-scale context with an expansive path to enable precise localization.

4. Adaptation of an end-to-end approach

In addition to the previously mentioned examples of end-to-end approaches, notable developments include the work by the Microsoft research group (Zhao et al., 2024) and Allen et al. (2025). These methods predict future atmospheric states on a regular grid directly from raw, heterogeneous observations. The core mechanism typically relies on a Vision Transformer (ViT; Dosovitskiy et al., 2021), structured into the following sequence of processing stages:

- **Encoding:** The impact of scattered observations is projected across the entire model domain;
- **Processing:** Latent representations and spatial relationships are propagated through self-attention layers;
- **Decoding:** A decoder, generally based on a convolutional or mesh-based architecture, projects latent states back to regular grids to produce atmospheric analyses or forecasts.

When applied to data assimilation (DA), this approach must be modified because the target is the present state (analysis) rather than a future state. A system demonstrating this strategy is FengWu-Adas (Chen et al., 2024), which adapts the FengWu ML emulator (Chen et al., 2023) to data assimilation. In global evaluations, it has been shown to outperform the physical Integrated Forecasting System (IFS) of the European Centre for Medium-Range Weather Forecasts (ECMWF) for medium-range forecasts beyond seven days.

When considering an end-to-end framework for RTMA, one concern is that spatial observation sparsity and limited high-frequency satellite imagery in certain regions could lead to an incomplete depiction of mesoscale dynamics. For this reason, the Aardvark Weather framework (Vaughan et al., 2024) may be particularly well suited for RTMA. In addition to sophisticated handling of irregular and non-stationary observations via multi-head self-attention mechanisms—which allow high computational parallelism—Aardvark naturally incorporates background trial fields and can be readily adapted to output real-time analyses instead of forecast trajectories.

The self-attention mechanism (Vaswani et al., 2017) was originally introduced as part of the Transformer architecture for Natural Language Processing (NLP) and subsequently generalized to spatial image processing. Attention facilitates cross-variable interactions and learns complex dependencies across disparate observation modalities. Transformers act as parallel context learners, mapping pairwise relationships across an entire sequence simultaneously (e.g., relating a specific token to all other tokens in a domain). This parallel processing makes Transformers computationally efficient and superior at capturing long-range spatial context compared to traditional sequential architectures like Recurrent Neural Networks (RNNs). The foundational engines of modern NLP, such as Generative Pretrained Transformers (GPT) and Bidirectional Encoder Representations from Transformers (BERT), rely on this core mechanism (shown schematically in Fig. 2).

A Vision Transformer (ViT) extends this framework to multi-dimensional visual and spatial data. Instead of utilizing traditional Convolutional Neural Networks (CNNs), ViTs divide input grids or images into a sequence of non-overlapping spatial patches—analogous to word tokens in text—and deploy self-attention mechanisms to learn spatial and physical relationships across these patches. Figure 3 provides a schematic overview of a ViT.

5. ML version of 4DVAR approach

Developing Tangent Linear (TL) and Adjoint (AD) representations of ML weather emulators (e.g., Tian et al., 2023) for variational data assimilation represents an attractive research avenue, primarily

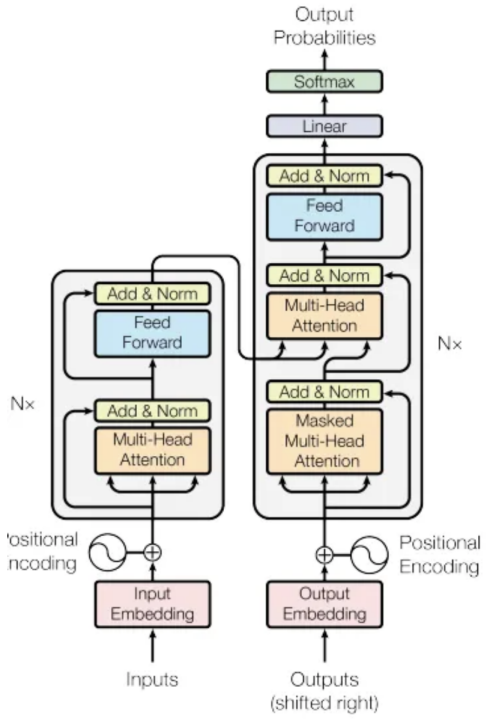


Figure 2: Schematic architecture of a Transformer model.

due to the speed with which ML surrogates can be constructed. Although Tian et al. (2024) raised concerns regarding the physical realism and numerical stability of purely data-driven ML TL and AD models, these issues can be addressed by hybridizing the ML emulator with physical equations incorporated as weak constraints in the training loss function. By penalizing violations of underlying dynamical balance equations, the loss function regularizes the network, encouraging generated state transitions to obey physical conservation laws alongside reference observations. Such physical regularization stabilizes the data-driven model, improving generalization when encountering out-of-distribution atmospheric states.

Alternatively, leveraging automatic differentiation frameworks (e.g., PyTorch Autograd) in ML architectures allows adjoint sensitivities to be computed seamlessly under the hood during the backward pass. This eliminates the need to manually derive and explicitly code separate TL and AD software modules within a 4D-Var framework—offering a promising avenue to streamline and accelerate future RTMA iterations.

In the following section, we illustrate these concepts using a simple shallow-water prototype.

6. Examples

The purpose of this section is to investigate, under simplified conditions, some of the methods discussed so far. We conduct a preliminary evaluation of the potential advantages of including physical constraints in ML emulators, and present several shallow-water prototypes of ML-based data assimilation, including a physically constrained ML 4D-Var.

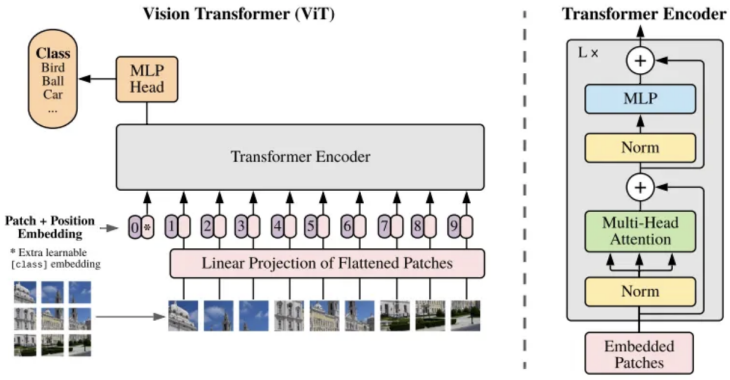


Figure 3: Schematic architecture of a Vision Transformer (ViT).

6.1 Inclusion of physics constraints in ML emulators

Physically Informed Neural Networks (PINNs) stand out as one of the robust frameworks for embedding physical laws directly into neural network architectures. PINNs were originally introduced by Raissi et al. (2017 a,b; 2019), and a comprehensive review can be found in Cuomo et al. (2022).

Including physics constraints in the ML weather forecasting model as an additional regularization of the loss function can stabilize the solution and extend its validity outside the range of situations in which the model was trained. To demonstrate potential benefit of physics constraints for variational assimilation, we conducted a 1D shallow water prototype experiment. Incorporating conservation laws into the neural network loss function significantly reduced unphysical drift and improved balance in the simulated fields. In the context of RTMA and 4D-Var, this approach offers two key operational advantages: first, it ensures that ML-based Tangent Linear (TL) and Adjoint (AD) surrogates generate physically consistent gradient updates without spurious noise; second, it constraints the analysis in observation-sparse regions where data alone cannot uniquely bound the solution. In an ML-based RTMA workflow, PINN architectures could be utilized to emulate rapid dynamic downscaling operators or non-linear surface-layer physical transformations. One can additionally include week balance constraints on the output from the neural network, to better prepare the analysis results to serve as initial conditions for a forecasting model (either ML based or physical).

We include here an example of inclusion of the physics constraints in a simple ML emulator that is solving a one-dimensional shallow-water model.

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - g\frac{\partial h}{\partial x} + \nu\frac{\partial^2 u}{\partial x^2} \quad , \tag{1}$$

$$\frac{\partial h}{\partial t} = -\frac{\partial uh}{\partial x} + \nu\frac{\partial^2 h}{\partial x^2} \quad . \tag{2}$$

The system is solved using central finite-differencing in space and the conventional fourth-order Runge-Kutta scheme (e.g., Johnson and Riess, 1982) for stepping in time, with periodic boundary conditions. The wind (u) and height (h) locations are split with respect to each other to prevent the generation of two-grid interval noise. The diffusion coefficient has the value $\nu = 0.001$, the number of points in the grid spread in the domain is $nx = 128$, and the time step is $dt = 0.001$ non-dimensional units. The model is run in the 'rollover' mode, in which the solution is estimated

at the end of 'large' time steps consisting of several 'small' time steps which satisfy the linear Courant-Friedrichs-Lewy (CFL) condition. We apply here a graph neural network (GNN), which, unlike fully connected standard neural networks, uses a 'message-passing' concept where nodes in the next layers are updated only from the immediate neighboring grid points.

We use winds, heights, and their discrete time derivatives $(\delta_t u, \delta_t h)$ at the input and output of the neural network. Time derivatives are defined from the finite difference form of the shallow-water equations that are being solved. Although this does not represent a typical application of PINNs, it should still give us an idea of the potential advantages of physical regularization of the loss function of the neural network. The whole system is first solved numerically and then this solution is used as a reference for training at the end of each of the rollover intervals. The model is being trained with 500 samples, defined as the sum of three waves whose parameters were randomly chosen for height and velocity.

Figure 4 shows the evolution of the loss function over 80 epochs in this "toy" shallow-water model constructed with the goal of demonstrating the effects of adding physics constraints to an ML emulator. In a series of validation tests, most of the samples show similar behavior of the solution, and here we show just one of them.

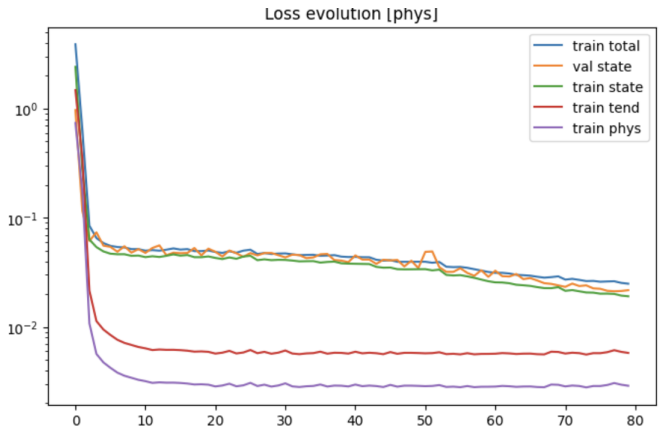


Figure 4: Evolution of loss function over 80 epochs.

Figures 5 show the evolution of the root mean square error (RMSE) of the chosen sample during the integration period. We see that the solution that is regularized by adding physics constraints is constantly lower (better) than the one without physics.

Figure 6 shows the evolution of the mass and energy derived from both the ML and the finite-difference (FD) integrations. Adding physics constraints generally leads to a more stable solution. In this particular case, adding the physics constraint helps with the conservation of total energy, defined as the sum of potential and kinetic energy.

Finally, in Fig. 7 we show the solution for height and wind, respectively, together with the final RMSE errors. Again, the benefits of adding physics constraints are clear.

We expect that by adding physics constraints in the full data-based emulator of the atmosphere, its TL and AD would similarly show a more physically realistic behavior, and such an approach could be used as a suitable variant of a ML 4DVAR version of RTMA. Furthermore, for RTMA specifically, PINN emulators can be applied to rapid high-resolution dynamic downscaling and surface-layer physics transformations, ensuring physical balance across complex terrain and observation-sparse domains.

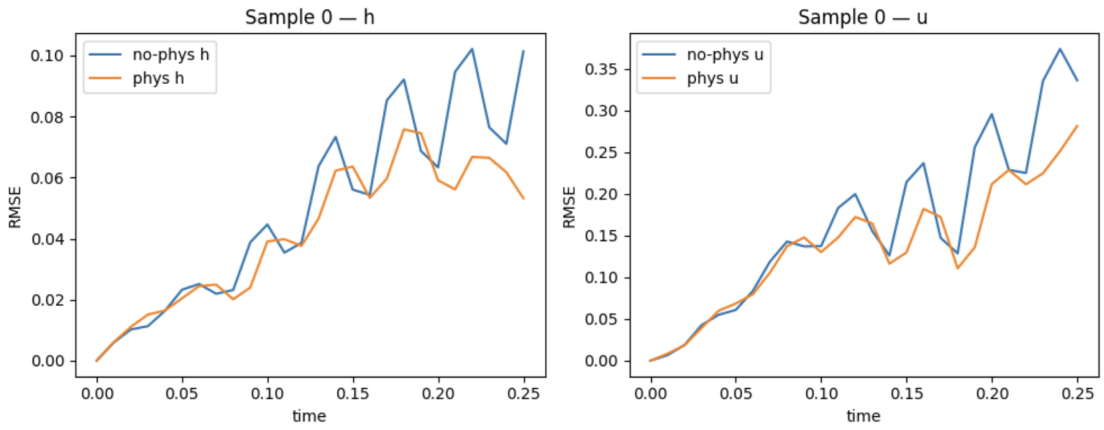


Figure 5: Evolution of height RMSE (left) and wind RMSE (right) over integration time across all rollover periods.

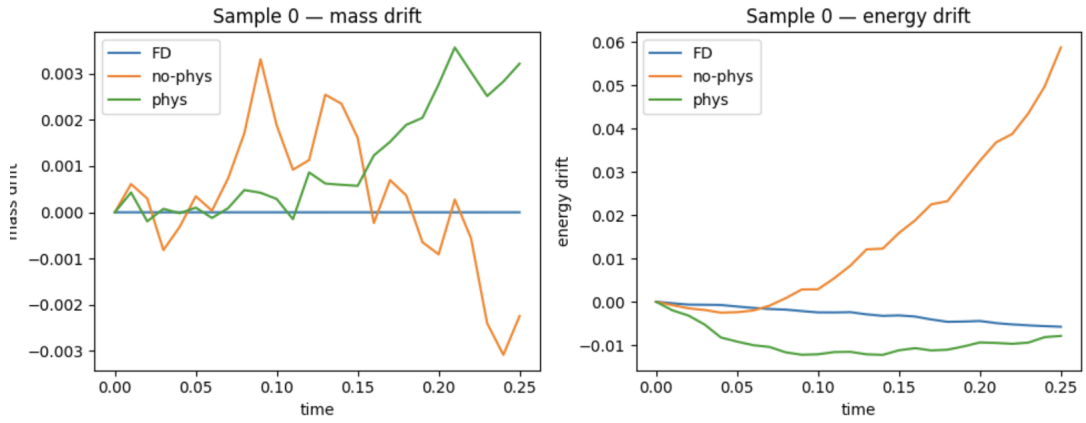


Figure 6: Evolution of mass (left) and total energy (right) over integration time across all rollover periods.

6.2 First ML prototype of data analysis for 2D RTMA

We constructed two prototype models to investigate how a simplified ML version of 2D RTMA could be designed to emulate an end-to-end data analysis workflow. Both prototypes employ a convolutional U-Net architecture.

In the first prototype, we define a 2D scalar field using a smooth analytical function, sampled across:

- A regular set of grid points representing the ”ground truth” (H);
- A randomly selected subset of grid points that represents synthetic observations.

We then slightly perturbed the original field to generate an ”imperfect background” (B) and map the scattered observations across all grid points of the domain to form an observation field (O). Next, we apply a convolutional U-Net configured with two input channels (B and O) and a single output channel ($\tilde{H}$). Mapping observations onto the model grid ensures the architecture can handle

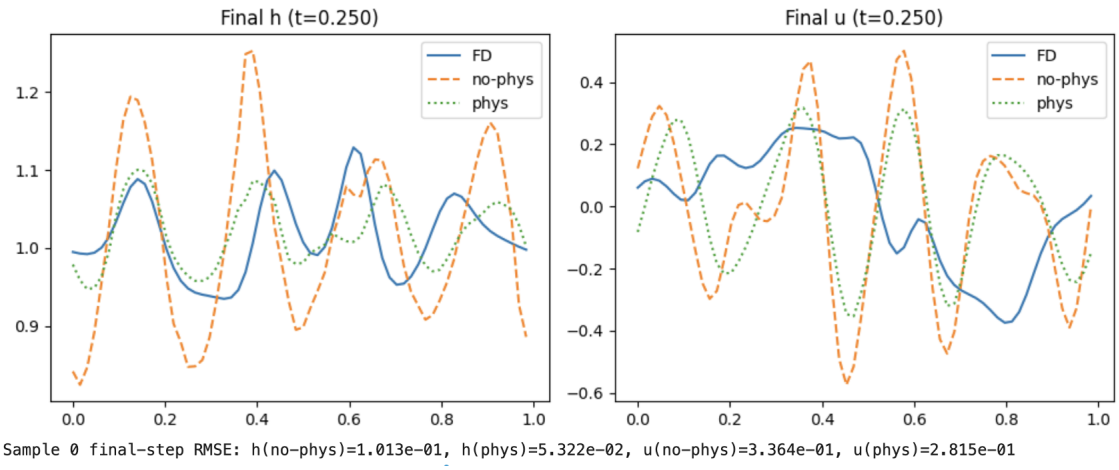


Figure 7: Solution of height (left) and wind (right) at the end of rollover periods.

missing or non-stationary observational data. The U-Net is trained across a series of realizations by evaluating the loss between the predicted output $\tilde{H}$ and the ground truth H.

Figure 8 shows the evolution of the training loss function for this first prototype test.

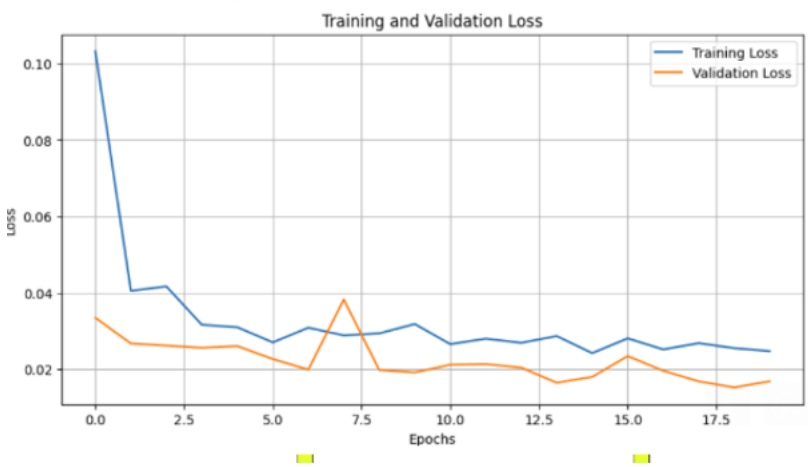


Figure 8: Evolution of the loss function for the first prototype ML data analysis model.

Figure 9 displays the background field and the mapped observations throughout the domain, while Figure 10 compares the target ground truth against the ML analysis output.

Although several open questions remain to be addressed through further experimentation, this preliminary exercise demonstrates a foundational step toward developing an ML-based data assimilation model.

6.3 Second ML prototype of data analysis for 2D RTMA

In the second prototype, we again employ the convolutional U-Net but expand the formulation to a multivariate state. We assume a shallow-water system governed by three primary state variables (u, v, h), representing the orthogonal velocity components and the height of the free-surface,

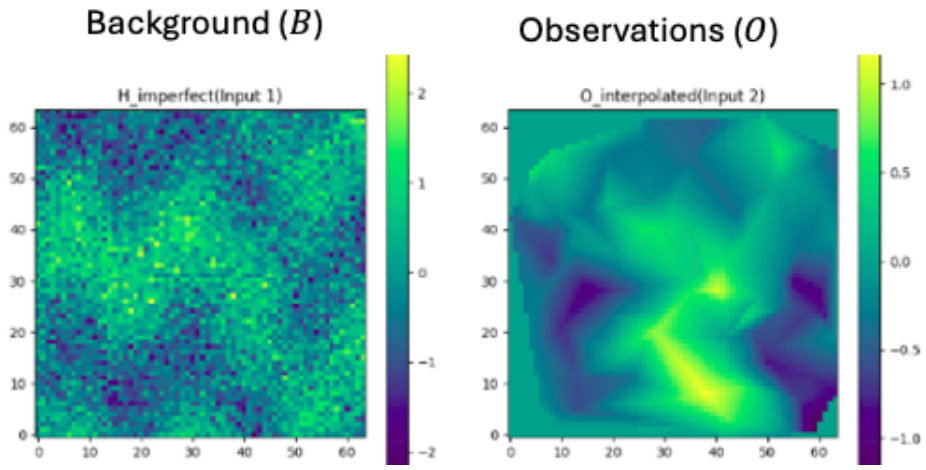


Figure 9: Input fields for the first prototype test: background field (left) and synthetic observations mapped across the domain (right).

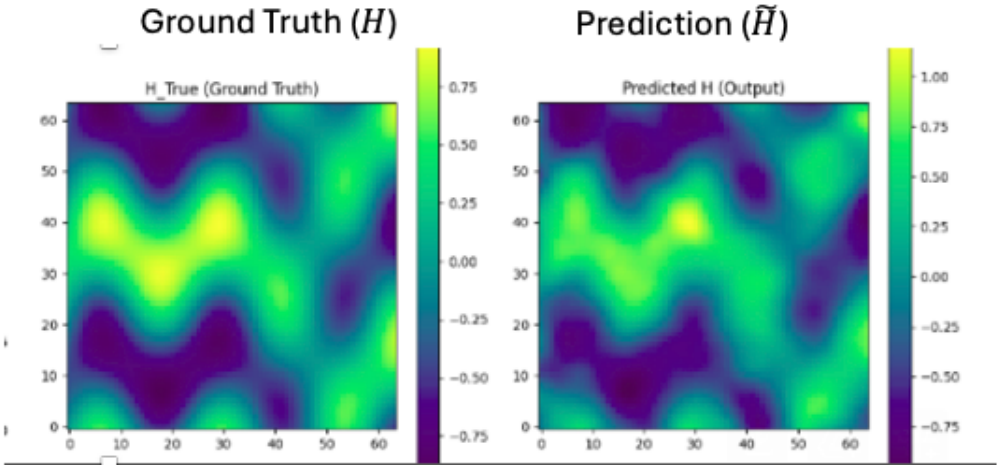


Figure 10: Ground truth target (left) and the resulting ML analysis field (right) for the first prototype test.

sampled at a set of randomly chosen grid locations acting as synthetic observations. In addition, we assume two sets of derived observation types (o_1, o_2), defined as simple non-linear functions of the state variables, also available at sparse, randomly distributed points. Small Gaussian errors are added to all synthetic observations before they are mapped onto the regular model grid via adjoint spatial interpolations to accommodate missing or non-stationary data. Consequently, the input to the neural network comprises five channels (u, v, h, o_1, o_2) and yields an output of three channels $(\tilde{u}, \tilde{v}, \tilde{h})$. Note that in this paradigm, the input observation mesh and output analysis grid do not strictly require identical spatial resolutions, although they are kept identical for this demonstration.

We executed a series of experiments at a relatively coarse spatial resolution to accelerate training iterations. The representative case presented here was conducted on a grid with $H = 256$ and $W = 512$ points. The height field in the benchmark state was specified analytically as:

$$h = \sin(X/2)\cos(Y/3) + 0.5\sin(2X/3 + Y/2) + \exp\left(-\frac{(X-2\pi)^2 + (Y-\pi)^2}{2(H/10)^2}\right). \quad (3)$$

Velocity fields were generated as perturbed geostrophic winds by adding random spatial noise to the geostrophic balance components followed by a smoothing filter.

Figure 11 shows the evolution of the loss function during training for this second prototype model.

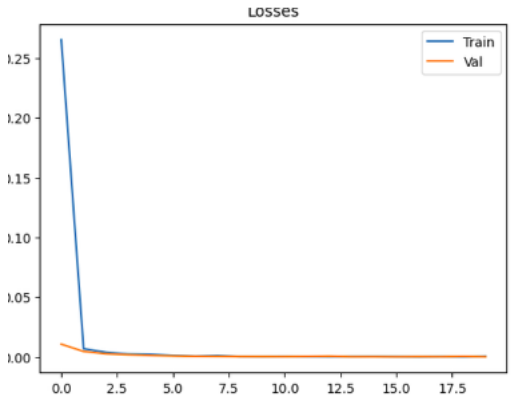


Figure 11: Evolution of the loss function for the second prototype ML data analysis model.

Figure 12 shows the spatial distribution of the normalized input synthetic observations before mapping to the regular grid.

Figure 13 compares the ground truth, the predicted ML analysis fields, and the resulting absolute error distributions.

Notably, this experiment operated without an explicit background trial field, relying solely on scattered observation inputs. Nevertheless, despite the added observational noise and the simple spatial mapping procedure, the reconstructed state fields exhibit strong agreement with the target truth.

As with the previous prototype, extensive further experimentation will be required to establish a rigorous scientific evaluation of this method. Nonetheless, these preliminary results demonstrate an encouraging step toward the development of an ML-based 2D RTMA framework.

6.4 PINNs-assisted 4DVAR

Using ML software frameworks allows us to construct a 4D-Var system without manually deriving and explicitly coding the tangent linear (TL) and adjoint (AD) models, as automatic differentiation automatically handles the required adjoint computations during the backward pass. Such an approach could serve as a pragmatic transitional step toward a fully ML-based data assimilation system. It enables leveraging much of the existing 3D RTMA infrastructure while improving analysis quality by accounting for the temporal distribution of observations within the assimilation window. The method applied here is referred to as ”PINNs-assisted 4D-Var” and shares similarities with the approach suggested by Purser (private communication), wherein the conjugate gradient solver in variational cost function minimization is replaced or augmented by a neural network.

We demonstrate the framework using a 1D shallow-water system, though the methodology readily generalizes to full atmospheric governing equations:

$$\frac{\partial \mathbf{q}}{\partial t} = -\frac{\partial \mathbf{F}}{\partial x} \quad , \tag{4}$$

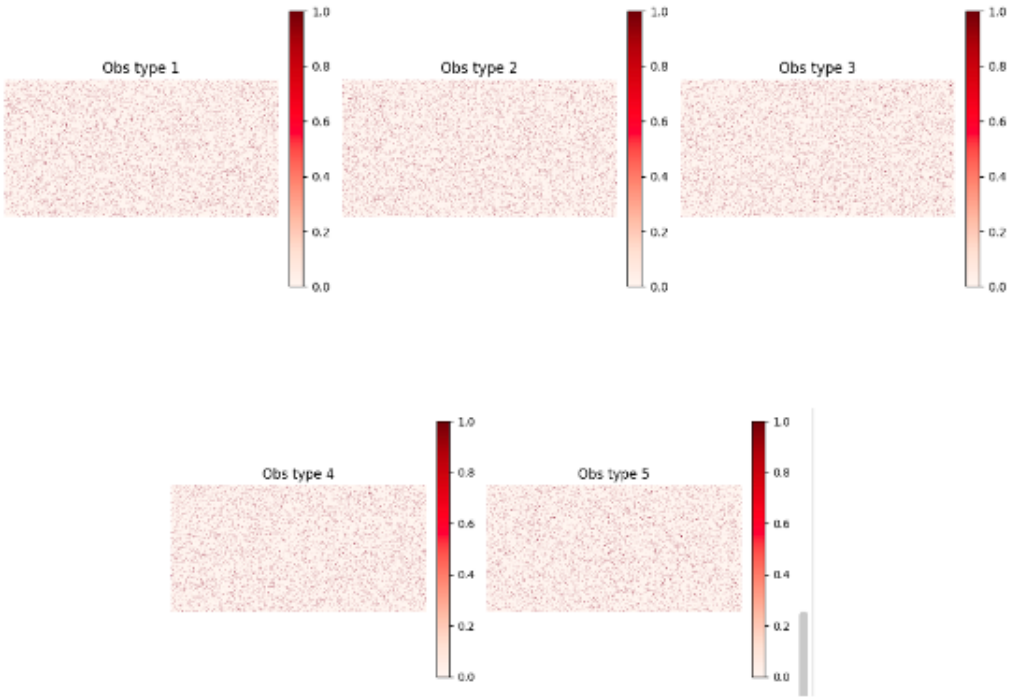


Figure 12: Normalized input synthetic observations for the second ML model prior to spatial grid mapping.

where

$$\mathbf{q} = \begin{bmatrix} h \\ u \end{bmatrix} \quad , \tag{5}$$

and

$$\mathbf{F} = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix} \quad . \tag{6}$$

Here, h, u, and g represent the fluid depth, flow velocity, and gravitational acceleration, respectively. The system is solved using the Rusanov (1961) scheme, a Godunov-type finite-volume method (e.g., Toro, 2013). The initial depth profile representing the ground truth is initialized as a sinusoidal wave in a domain with periodic boundary conditions (Figure 14). The background field is generated by adding smooth perturbations to the truth, constructed by applying a 3-point moving average filter three times to white noise (equivalent to repeatedly convolving with a compact smoothing kernel). The truth state is integrated forward in time, and synthetic observations ("Obs") are produced by corrupting a subset of true state values with Gaussian noise. During optimization, compact Convolutional Neural Networks (CNNs) learn residual differences between predicted and observed states, outputting conservative flux corrections that incrementally nudge the analysis state toward balance.

The central advantage of this architecture is that, rather than writing a dedicated adjoint model, backward sensitivity passes are managed transparently by PyTorch automatic differentiation because every step in the forward integration (finite-difference solver, flux calculation, divergence evaluation, and neural network operations) consists of differentiable tensor operations. Furthermore, within this weak-constraint 4D-Var formulation, model error is represented directly by the CNN flux corrections.

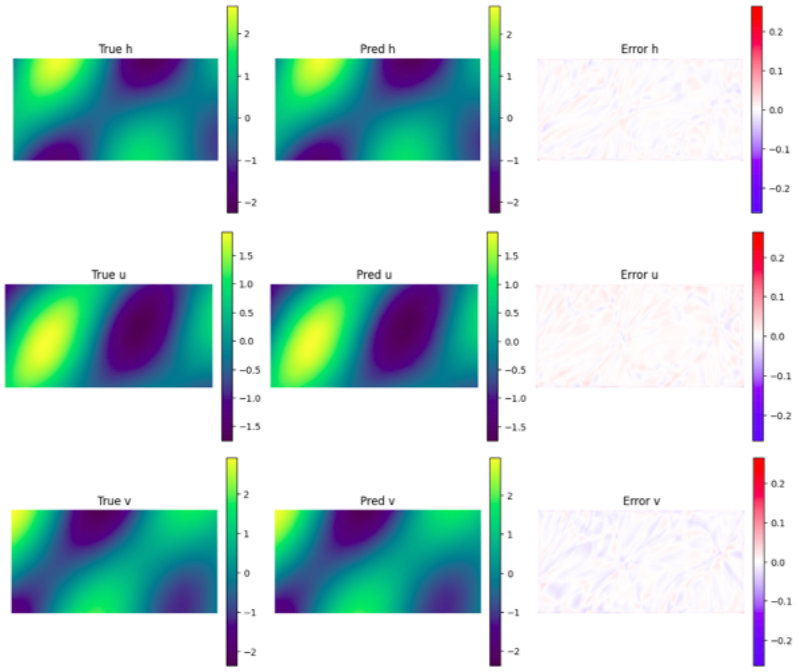


Figure 13: Ground truth target (left column), predicted ML analysis fields (middle column), and spatial error distributions (right column) for the second prototype model.

The key results of these tests are summarized in Figures 15 and 16, demonstrating that the resulting analysis state consistently outperforms the background trial field relative to the truth.

This strategy can be combined with any physical numerical model and, with minor extensions, can also be integrated alongside ML weather emulators, provided that the emulator architecture supports differentiable tensor operations.

7. Conclusions

The primary purpose of this text is to present an initial set of explorations and to stimulate further research and discussion regarding the application of AI/ML in data assimilation. We provide an overview of emerging strategies for integrating machine learning into meteorological data analysis, with a particular focus on advancing the RTMA project. Our exploration of several prototype models and the potential to exploit 4D-Var without the need for explicitly coded TL and AD models underscores the rapid evolution of this field. The breadth of possible ML applications to data assimilation (DA) continues to expand, and we recognize that several promising avenues were not exhaustively covered. For example, Graph Attention Networks (GATs; Veličković et al., 2018) offer unique advantages for processing irregularly distributed observation datasets—a capability highly relevant to atmospheric data assimilation that warrants deeper investigation.

We suggest the application of PINNs as a physically consistent mechanism to extend the validity of ML emulators to out-of-distribution regimes, such as extreme weather phenomena missing from training datasets. This capability is increasingly critical in an era of accelerating climate change. Within data assimilation specifically, incorporating PINNs can enhance the physical realism and numerical stability of TL and AD operators derived from ML surrogates.

As our preliminary tests demonstrate, the PINNs-assisted 4D-Var (e.g., Krueger and Ward,

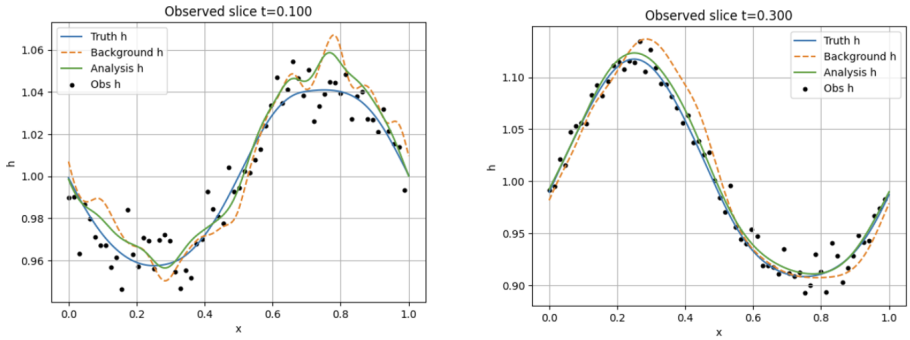


Figure 14: Cross-section through two non-dimensional times (0.1 and 0.3) in a preliminary test of the PINNs-assisted 4D-Var method with a 1D shallow-water model. The initial height field representing the truth is sinusoidal. The background is generated via smooth random perturbations of the truth, and synthetic observations are generated by corrupting true values with Gaussian noise. Both truth and background states are propagated forward with a finite-volume solver. The analysis is computed via CNN flux corrections, while backward sensitivities are evaluated using PyTorch automatic differentiation.

2025) offers an attractive ML-based DA framework capable of correctly processing observations distributed continuously across the assimilation window. In early operational phases, this approach could introduce a temporal dimension to 2D RTMA while preserving the majority of the existing infrastructure. This line of research closely aligns with the ”Neural GCM” paradigm developed by Google Research (Kochkov et al., 2024)—a hybrid architecture wherein a differentiable dynamical core governs large-scale fluid dynamics using classical numerical schemes, while neural networks parameterize subgrid-scale physical processes (e.g., clouds, convection, radiation). Because the entire system is end-to-end differentiable, variational optimization (akin to 4D-Var) can be executed directly across the combined model.

Machine learning also presents new opportunities for Ensemble Kalman Filtering (EnKF). Leveraging large ensembles generated rapidly by ML emulators, such as those described by Weyn et al. (2021), opens new avenues for ensemble-based data assimilation. Our ongoing research

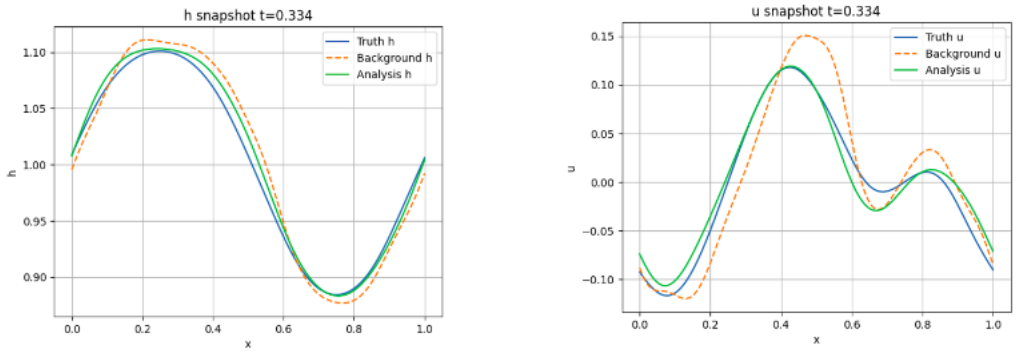


Figure 15: Snapshot of the analysis, background, and truth states after 0.334 non-dimensional time units in the preliminary test of the PINNs-assisted 4D-Var framework.

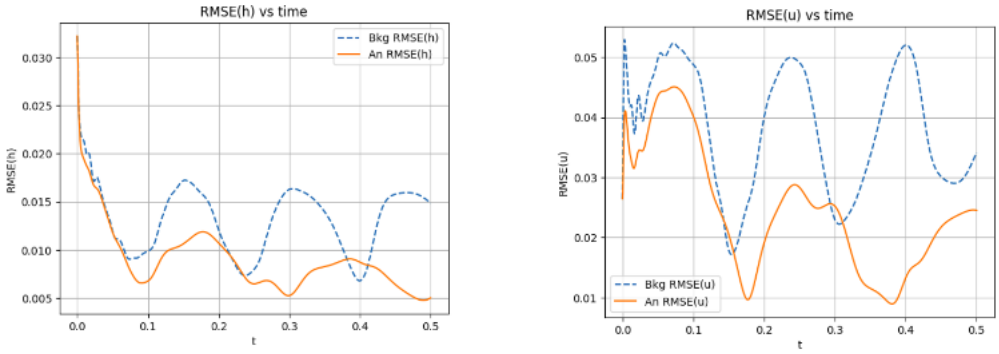


Figure 16: Temporal evolution of root mean square error (RMSE) in the preliminary test of the PINNs-assisted 4D-Var framework.

into dynamic-stochastic modeling via ML aims to streamline this process further by replacing large ensemble explicit runs with a single probabilistic model—promising substantial computational speedups while maintaining the statistical rigor required for high-resolution analyses.

Graph Neural Networks (GNNs) and CNNs can also be deployed to formulate flow-dependent adaptive localization functions for ensemble covariance modeling (e.g., Howard et al., 2024; Kotsuki et al., 2025). Additional AI applications include estimating analysis error variance (e.g., Farchi et al., 2024). Furthermore, Bocquet et al. (2025) demonstrate that Generative AI and diffusion models can quantify analysis uncertainty by sampling synthetic ensembles that represent true non-Gaussian probability density functions more accurately than traditional Gaussian approximations.

A central challenge in applying ML to DA is managing observations that are irregularly distributed in space and time or are periodically missing. The Aardvark framework addresses this challenge using multi-head self-attention mechanisms embedded within Vision Transformers. In our U-Net prototype experiments, we addressed observation irregular grid alignment by mapping observations onto surrounding grid points using bilinear adjoint interpolation. However, a more mathematically rigorous approach involves Radial Basis Functions (RBFs; e.g., Wendland, 2005; Rancic, 2026), which are designed specifically for high-fidelity scattered data interpolation. Figure 17 illustrates an example mapping 70 sparse randomly distributed observations onto a regular evaluation grid using a hierarchical two-stage RBF pipeline. In the first stage, a global Multiquadric RBF equipped with physics-aware analytical boundary reflection (enforcing parity across domain limits) projects the observations onto an intermediate synthetic control mesh. A localized Gaussian RBF kernel then refines the background field, ensuring a smooth, physically realistic spatial spread of observational influence. Employing compact-support bump or beta kernels could offer further computational efficiency in real-time assimilation steps. It would also be of significant interest to compare the performance and spatial covariance representation of this RBF embedding framework against modern attention- or graph-based learning architectures within ML data assimilation workflows.

Regarding computational implementation, several observations should be made regarding programming language selection for ML RTMA. While Python is widely viewed as the dominant language for GPU-accelerated computing, Fortran remains central to operational meteorology due to its computational speed, numerical stability, parallel execution capabilities, and decades of validated production code. Extensions enabling Fortran to harness GPU acceleration include CUDA Fortran (developed via NVIDIA NVFORTRAN compilers) and OpenACC directives. Given

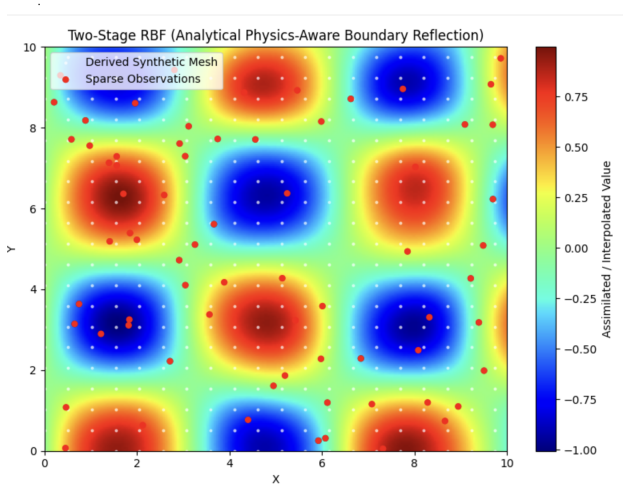


Figure 17: Reconstruction of an unaligned 2D field from 70 sparse observations using a two-stage hierarchical Radial Basis Function (RBF) pipeline. The framework employs a global Multiquadric RBF with physics-aware boundary reflection to derive an intermediate synthetic control mesh (white grid), followed by a localized Gaussian kernel to yield a continuous, edge-stable state field.

the extensive existing Fortran infrastructure supporting RTMA, leveraging these GPU directives offers a practical pathway to modernize the codebase without requiring a complete rewrite from scratch or total reliance on external Python frameworks. Meanwhile, emerging languages such as Julia (combining C-like speed with Python-like ease) and Rust (optimized for high-performance data infrastructure) are gaining traction in scientific ML. Additionally, Mojo—developed by Modular, Inc.—combines Python syntax with system-level performance (offering compilation speedups up to several thousand times faster than standard Python for specialized numerical tasks), allowing direct interoperability with existing Python libraries.

Looking ahead, it is essential to continue exploring hybrid ML architectures that incorporate domain knowledge and physical balance constraints. Interdisciplinary collaboration between meteorologists and machine learning specialists will be key to unlocking the potential of these tools in operational weather prediction. As a practical first step toward an ML 3D RTMA, leveraging accessible frameworks like the Vision Transformer (ViT) architecture used in Aardvark represents a promising direction, given its inherent ability to process non-stationary observations within the assimilation window. Although Aardvark is an end-to-end system designed for unified analysis and forecasting, our comparative evaluations indicate that lightweight U-Net architectures can produce comparable state reconstructions under specific operational constraints. Another attractive alternative is to try to adjust framework of OCELOT (Gholoubi et al. 2026), ML prediction model that works fully in observational space, which would allow to streamline development efforts. In the long term, adapting a diffusion-based framework (e.g. DiffDA) may represent the optimal destination. Generative diffusion models introduce intrinsic probabilistic capabilities, excel at ultra-high-resolution analysis generation, and dramatically reduce inference execution times, aligning directly with the operational objectives of the RTMA project.

In the end, we outline the possible major steps toward developing an operational ML-based RTMA framework capable of leveraging these emerging methodologies. We also highlight key research challenges that must be addressed throughout the development process:

1. **Develop an ML 3D-Var prototype using Aardvark and/or OCELOT infrastructure, or build a bespoke ML 3D-Var architecture from scratch:**
    - Investigate methods for embedding dynamical constraints into loss functions, such as weak geostrophic balance and energy conservation;
    - Formulate strategies that enable the ML analysis to effectively learn flow-dependent atmospheric error covariances directly from observation networks.
2. **Develop an ML 4D-Var framework for RTMA:**
    - Within Aardvark or a custom framework, replace the classical numerical model with a differentiable ML emulator and evaluate performance gains when incorporating PINNs;
    - Within OCELOT, construct an ML 4D-Var operating directly in observation space as an AI counterpart to the representer method (e.g., Bennett, 2002).
3. **Develop a DiffDA-based architecture to incorporate probabilistic capabilities:**
    - Formulate generative diffusion models to provide ensemble analysis distributions and quantify non-Gaussian analysis uncertainties at high spatial resolutions.
4. **Expand observation ingestion capability and targeted mesoscale parameters:**
    - Leverage multi-source observational datasets, focusing specifically on improving challenging diagnostic fields such as cloud ceiling height and surface visibility.

8. References

Abdi, D., I. Jankov, P. Madden, V. Vargas, T. A. Smith, S. Frolov, M. Flora, and C. Potvin, 2025: HRRRCast: A data-driven emulator for regional weather forecasting at convection allowing scales. arXiv:2507.05658.

Allen, A., and co-authors, 2025: End-to-end-driven weather prediction. Nature, 641 (8065), 1172–1179, https://doi.org/10.1038/s41586-025-08897-0.

beginhangparas1.5em1 Bocquet, M., 2025: Machine learning for data assimilation: Blitz literature overview and classification. ECMWF Annual Seminar.

Bennett, A. F., 2002: Inverse Modeling of the Ocean and Atmosphere. Cambridge University Press, 234 pp.

Bi, K., and co-authors, 2023: Accurate medium-range global weather forecasting with 3D neural networks. Nature, 619, 533–538, https://doi.org/10.1038/s41586-023-06185-3.

Bonavita, M., and P. Laloyaux, 2020: Machine learning for model error inference and correction. Journal of Advances in Modeling Earth Systems, 12 (12), e2020MS002232, https://doi.org/10.1029/2020MS0022

Brock, F. V., K. C. Crawford, R. L. Elliott, G. W. Cuperus, S. J. Stadler, H. L. Johnson, and M. D. Eilts, 1995: The Oklahoma Mesonet: A Technical Overview. Journal of Atmospheric and Oceanic Technology, 12 (1), 5–19.

Chen, K., and co-authors, 2023: Fengwu: Pushing the skillful global medium-range weather forecast beyond 10 days lead. arXiv:2304.02948.

Chen, K., and co-authors, 2024: Towards an end-to-end artificial intelligence driven global weather forecasting system. arXiv:2312.12462.

Cuomo, S., V. S. Di Cola, F. Giampaolo, G. Rozza, M. Raissi, and F. Piccialli, 2022: Scientific machine learning through physics-informed neural networks: Where we are and whats next. Journal of Scientific Computing, 92 (3), 88, https://doi.org/10.1007/s10915-022-01939-z.

De Pondeca, M. S., and co-authors, 2011: The real-time mesoscale analysis at NOAA's National Centers for Environmental Prediction: Current status and development. Weather and Forecasting, 26 (5), 593–612, https://doi.org/10.1175/WAF-D-10-05037.1.

DesRosiers, A. J., and M. M. Bell, 2024: Airborne radar quality control with Machine Learning. Artificial Intelligence for the Earth Systems, 3 (1), e230043, https://doi.org/10.1175/AIES-D-23-0043.1.

Dosovitskiy, A., L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, 2021: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929.

Du, H., and co-authors, 2025: WeatherMesh-3: Fast and accurate operational global weather forecasting model. arXiv:2503.22235.

Farchi, A., P. Laloyaux, M. Bonavita, and M. Bocquet, 2021: Using machine learning to correct model error in data assimilation and forecast applications. Quarterly Journal of the Royal Meteorological Society, 147 (739), 3067–3084, https://doi.org/10.1002/qj.4116.

Ge, G., and co-authors, 2024: The 3D Real-Time Mesoscale Analysis (3DRTMA) for severe weather, aviation, operational forecasting, and other nowcast applications. The 10th International Symposium on Data Assimilation (ISDA2024), Kobe, Japan.

Gholoubi, A., R. McLaren, M.-C. Ko, X. Jin, N. Esposito, N. Collard, C. Martin, R. Treadon, D. Holdaway, and D. Kleist, 2026: OCELOT: Direct Atmospheric Forecasting from Heterogeneous Earth Observations Using a Graph-Transformer Hybrid Model. arXiv:2607.14196.

Gyawali, D., 2023: Comparative analysis of CPU and GPU profiling for deep learning models. arXiv:2309.02521.

Hodyss, D., and M. Morzfeld, 2025: Using diffusion models to do data assimilation. arXiv:2506.02249.

Howard, L. J., A. Subramanian, and I. Hoteit, 2024: A machine learning augmented data assimilation method for high-resolution observations. Journal of Advances in Modeling Earth Systems, 16 (6), e2023MS003774, https://doi.org/10.1029/2023MS003774.

Huang, L., L. Gianinazzi, Y. Yuejiang, P. D. Dueben, and T. Hoefler, 2024: DiffDA: A diffusion model for weather-scale data assimilation. arXiv:2401.05932.

Johnson, L. W., and R. D. Riess, 1982: Numerical Analysis, 2nd edition. Addison-Wesley, Reading MA, 563 pp.

Kochkov, D., and co-authors, 2024: Neural general circulation models for weather and climate. Nature, 632, 1060–1067, https://doi.org/10.1038/s41586-024-07744-y.

Kotsuki, S., K. Shiraishi, and A. Okazaki, 2025: Ensemble data assimilation to diagnose AI-based weather prediction models: a case with ClimaX version 0.3.1. Geoscientific Model Development, 18 (20), 7215–7225, https://doi.org/10.5194/gmd-18-7215-2025.

Krueger, N. N., and J. T. Ward, 2025: Reparameterizing 4DVAR with neural fields. arXiv:2509.21751.

Lam, R., and co-authors, 2023: Learning skillful medium-range global weather forecasting. Science, 382 (6677), 1416–1421, https://doi.org/10.1126/science.adi2336.

Larsson, E., J. Oskarsson, T. Landelius, and F. Lindsten, 2025: Diffusion-LAM: Probabilistic limited-area weather forecasting with diffusion. arXiv:2502.10234.

Ma, X., X. Dong, A. Tarrant, L. Yang, R. Kotamarthi, J. Wang, F. Yan, and R. Kettimuthu, 2025: Diffusion-based, data-assimilation-enabled super-resolution of hub-height winds. arXiv:2510.03364.

Manshausen, P., and co-authors, 2025: Generative data assimilation of sparse weather station observations at kilometer scales. arXiv:2406.16947.

McPherson, R. A., and co-authors, 2007: Statewide monitoring of the mesoscale environment: A technical update on the Oklahoma Mesonet. Journal of Atmospheric and Oceanic Technology, 24 (3), 301–321.

Oskarsson, J., T. Landelius, and F. Lindsten, 2023: Graph-based neural weather prediction for limited area modeling. arXiv:2309.17370.

Pathak, J., and co-authors, 2022: FourCastNet: A global data-driven high-resolution weather model using adaptive Fourier neural operators. arXiv:2202.11214.

Pathak, J., and co-authors, 2024: Kilometer-scale convection allowing model emulation using generative diffusion modeling. arXiv:2408.10958.

Pathak, J., and co-authors, 2026: Learning accurate storm-scale evolution from observations. NVIDIA Research, https://research.nvidia.com/publication/2026-01$_{l}earning-accurate-storm-scale-evolution-observations$.

Raissi, M., P. Perdikaris, and G. E. Karniadakis, 2017a: Physics informed deep learning (Part I): Data-driven solutions of nonlinear differential equations. arXiv:1711.10561.

Raissi, M., P. Perdikaris, and G. E. Karniadakis, 2017b: Physics informed deep learning (Part II): Data-driven discovery of nonlinear differential equations. arXiv:1711.10566.

Raissi, M., P. Perdikaris, and G. E. Karniadakis, 2019: Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics, 378, 686–707.

Rancic, M., 2026: Reconstruction of frontal gradients using Radial Basis Function interpolation. Atmosphere, 17 (8), 718, https://doi.org/10.3390/atmos17080718.

Rombach, R., A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, 2022: High-resolution image synthesis with latent diffusion models. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 10684–10695.

Rusanov, V. V., 1961: The calculation of the interaction of non-stationary shock waves with obstacles. USSR Computational Mathematics and Mathematical Physics, 1 (2), 267–279 (1962 English Translation).

Sadeghi Tabas, S., J. Wang, W. Li, M. Row, Z. Zhang, L. Zhu, J. Peng, and J. R. Carley, 2025: GFS-powered machine learning weather prediction: A comparative study on training GraphCast with NOAA's GDAS data for global weather forecast. NCEP Office Note 521, https://doi.org/10.25293/xd3y-wy31.

Selmer, P., J. E. Yorks, E. P. Nowottnick, A. Cresanti, and K. E. Christian, 2024: A deep learning lidar denoising approach for improving atmospheric feature detection. Remote Sensing, 16 (15), 2735, https://doi.org/10.3390/rs16152735.

Shen, Z., 2025: Conditional denoising score matching for sequential data assimilation. Ocean-Land-Atmosphere Research, 4, Article 0103, https://doi.org/10.34133/olar.0103.

Sun, M., B. Zhang, J. Sun, and K. Gao, 2021: Deep-learning-based precipitation observation quality control. Journal of Atmospheric and Oceanic Technology, 38 (5), 897–911.

Tian, X., L. Conibear, and J. Steward, 2023: A neural-network based MPAS-shallow water model and its 4D-Var data assimilation system. Atmosphere, 14 (1), 157, https://doi.org/10.3390/atmos14010157.

Tian, X., D. Holdaway, and D. Kleist, 2024: Exploring the use of machine learning weather models in data assimilation. arXiv:2411.14677.

Toro, E. F., 2013: Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction. Springer Science & Business Media, 724 pp.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, 2017: Attention is all you need. Advances in Neural Information Processing Systems (NeurIPS 2017), 30, 5998–6008.

Vaughan, A., and co-authors, 2024: End-to-end data driven weather prediction. arXiv:2404.00411.

Veličković, P., G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, 2018: Graph attention networks. International Conference on Learning Representations (ICLR 2018), https://arxiv.org/abs/1710.10903.

Virmani, V., V. Ranga, and G. Virmani, 2024: AI and machine learning in climate change research: A review of predictive models and environmental impact. World Journal of Advanced Research and Reviews, 21 (3), 011–020.

Wendland, H., 2005: Scattered Data Approximation. Cambridge University Press, 348 pp.

Weyn, J. A., D. R. Durran, R. Caruana, and N. Cresswell-Clay, 2021: Sub-seasonal forecasting with a large-ensemble of deep-learning weather prediction models. Journal of Advances in Modeling Earth Systems, 13 (7), e2021MS002502, https://doi.org/10.1029/2021MS002502.

Yang, L., and co-authors, 2025: Diffusion models: A comprehensive survey of methods and applications. arXiv:2209.00796.

Zhao, P., J. Bian, Z. Ni, W. Jin, J. Weyn, Z. Fang, S. Xiang, H. Dong, B. Zhang, H. Sun, K. Thambiratnam, and Q. Zhang, 2024: OMG-HD: A high-resolution AI weather model for end-to-end forecasts from observations. arXiv:2412.18239.

Zhuang, F., and co-authors, 2021: A comprehensive survey on transfer learning. Proceedings of the IEEE, 109 (1), 43–76, https://doi.org/10.1109/JPROC.2020.3004555.

APPENDICES

A. Glossary of acronyms used

4DVAR Four-dimensional Variational Data Assimilation

AI/ML Artificial Intelligence and Machine Learning

AD Adjoint

BERT Bidirectional Encoder Representations of Transformers

CFL Courant-Friedrichs-Lewy

CNN Convolutional Neural Network

CONUS Contiguous United States domain

CPU Central Processing Unit

CUDA Compute Unified Device Architecture

DA Data Assimilation

DiffDA Diffusion model for data assimilation

ECMWF European Centre for Medium-Range Weather Forecasts

FAA Federal Aviation Administration

GAT Graph Attention Network

GCM General Circulation Model

GNN Graph Neural Network

GPT Generative Pretrained Transformer

GSL Global Systems Laboratory

IFS Integrated Forecasting System

MDC Modeling and Development Center

NWS National Weather Service

NLP Natural Language Processing

PINN Physically Informed Neural Network

OMD Office of Modeling and Development

RMSE Root Mean Square Error

RBF Radial Base Function

RTMA Real-Time Mesoscale Analysis

TL Tangent Linear

TML Transfer Machine Learning

URMA Unrestricted Mesoscale Analysis

ViT Visual Transformer